REMARKS

The foregoing amendment and the following arguments are provided to impart precision to the claims, by more particularly pointing out the invention, rather than to avoid prior art.

The Examiner has objected to the drawings.

35 U.S.C. § 102(e) Rejections

Examiner rejected claims 1-3, 5-12, 14-25 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent Number 6,091,897 (hereinafter "Yates").

> "To anticipate a claims, the reference must teach every element of the claim. A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." (Manual of Patent Examining Procedures (MPEP) ¶ 2131.)

Independent claims 1, 10, and 19 of the present application include limitations not disclosed or taught by Yates. As a result, claims 1, 10, and 19 are not anticipated by Yates.

In particular, the independent claims ?, include the limitation, or a limitation similar thereto, of testing to see if resources needed by a *block of code* are available. The limitation of testing to see if resources needed by a *block of code* are available is not disclosed nor suggested by Yates.

Rather, Yates discloses processing on an instruction by instruction basis: "The operating system initiates the execution of an instruction conversion

program 16 or feeds the file *instruction by instruction* to an instruction pre-processor." (Yates, col. 9, lns. 25-28) (emphasis added).

Furthermore, Yates does not disclose the claimed limitation of *testing to see if the resources are available.* Rather, Yates discloses determining if a previously generated binary translation is available for the code that is currently requested to be translated:

> Prior to performing a conversion by the run time system 32, the run-time system 32 interrogates the server process 36 via a path 32a to determine from the server process *whether there is a native image corresponding to the routine of the application program stored in segment 17b whose execution has just been requested by a user*. (Yates, col. 10, lns. 15-28) (emphasis added).

As a result of Yates not disclosing nor suggesting limitations included in applicant's independent claims, Yates does not anticipate independent claims 1, 10, and 19.

In addition, the remaining claims depend from one of independent claims 1, 10, and 19, and therefore include the distinguishing claim limitations of claims 1, 10, and 19, as discussed above. As a result, the remaining claims are also not anticipated by Yates.

CONCLUSION

Applicants respectfully submit the present application is in condition for allowance. If the Examiner believes a telephone conference would expedite or assist in the allowance of the present application, the Examiner is invited to call John Ward at (408) 720-8300, x237.

Authorization is hereby given to charge our Deposit Account No. 02-2666 for any charges that may be due.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Date: 03/10/2003

John P. Ward
Reg. No. 40,216

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026
(408) 720-8300

ATTACHMENT A

A marked-up version of the amended specification is as follows:

IN THE SPECIFICATION:

On page 6, please replace the paragraph beginning at line 3 with the following rewritten paragraph:

- - (Amended) A method and apparatus for emulating a computer processor architectural stack is disclosed. In the following description, for purposes of explanation, specific nomenclature is set forth to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. For example, the present invention has been described with reference to the [Intel] INTEL architecture floating point stack. However, the same techniques can easily be applied to other types of stacks in other computer processors. - -

On page 7, please replace the paragraph beginning at line 5 with the following rewritten paragraph:

- - (Amended) To illustrate how prior art software emulators and binary translator systems handle exceptions, an example is provided. The following example illustrates the traditional approach for validating stack limitations. The following code lists a short program for 32-bit [Intel] INTEL architecture processors. - -

On page 7, please replace the paragraph beginning at line 10 with the following rewritten paragraph:

- - (Amended) The first line of the 32-bit [Intel] INTEL architecture pushes a floating point value from a memory location <mem> onto the processor's floating point stack. The second line of code pops a value off the processor's floating point stack and puts the value into a destination memory location <dest1>. The third line of code pops a value off the processor's floating point stack and puts the value into a destination memory location <dest2>. - -